Why you need to use this ?

1. you do not need to create any controller files. because system will create a controller for you while you create the menu using the interface.

2. you do not need to create a separate modules (Eg for sub module :- "Manage users" ), because the system will create a separate module while you create the sub menu.

3. you do not need to create any JS,CSS files for you new module. because the system will automatically create it for you.

4. you do not need to create any functions in the controller. because the system will create for you when you create the sub menu function.

\*so developers no need to configure the navigation bar , controller, model , views and module functions.

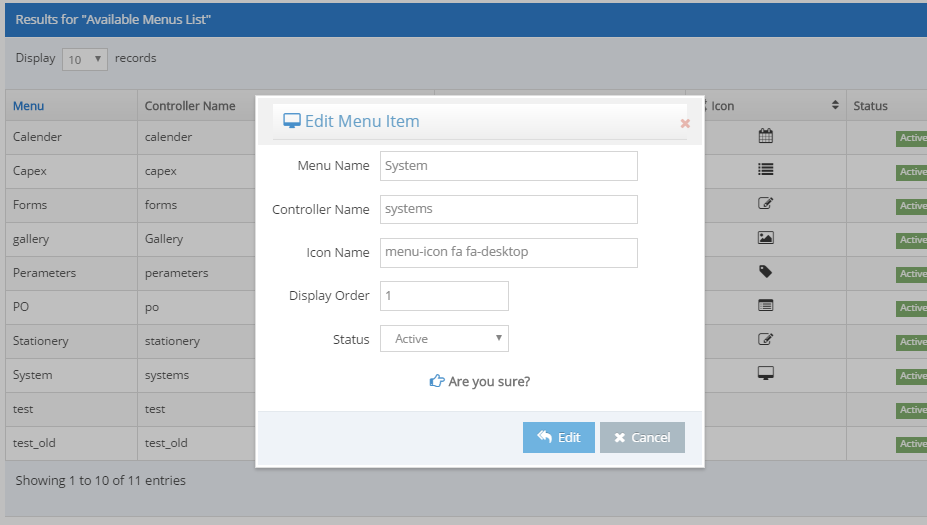
\*they can directly start the coding by accessing the function quickly. it will minimize the developing time.

5. this system having a user management system. so you do not need to waste your time to develop a separate user management to this system.

so all you need to do is just login to the system as admin(username: admin, password:admin123) the system interface . and then,

a. create the menu

steps=> click "system" nav => click "manage menus" => click "add menu" button => fill the data and submit.

Eg :- new controller is "test"

so according to the example,

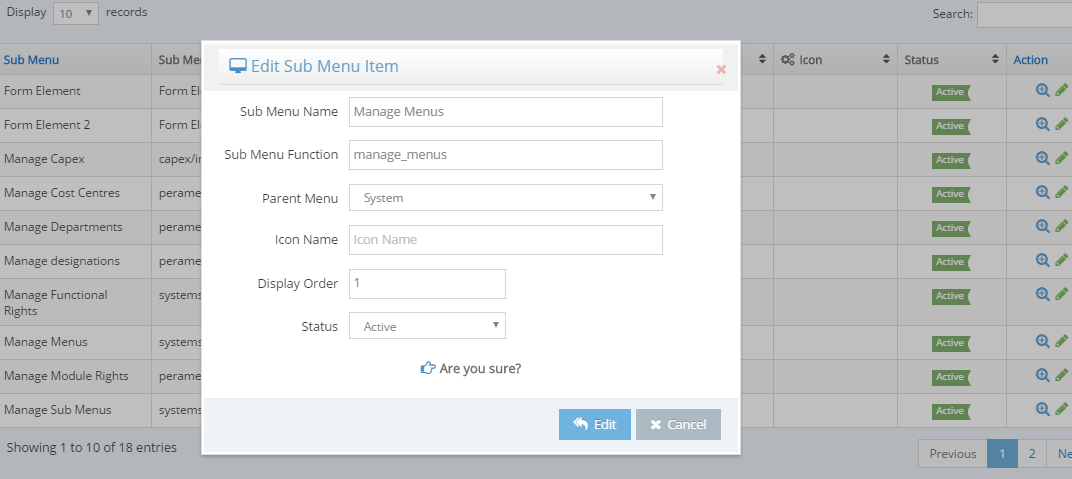
your controller name : test

your model name : tes

your view folder : test

b. Create sub menu

steps => click "system" nav => click "manage sub menus" => click "add sub menu" button => fill the data and submit.

Eg :- new medule is "test"

so according to the example,

your new module : "Manage Menus"

your new module function : manage\_menus (this is the main link to this module). you can edit this function through controller.

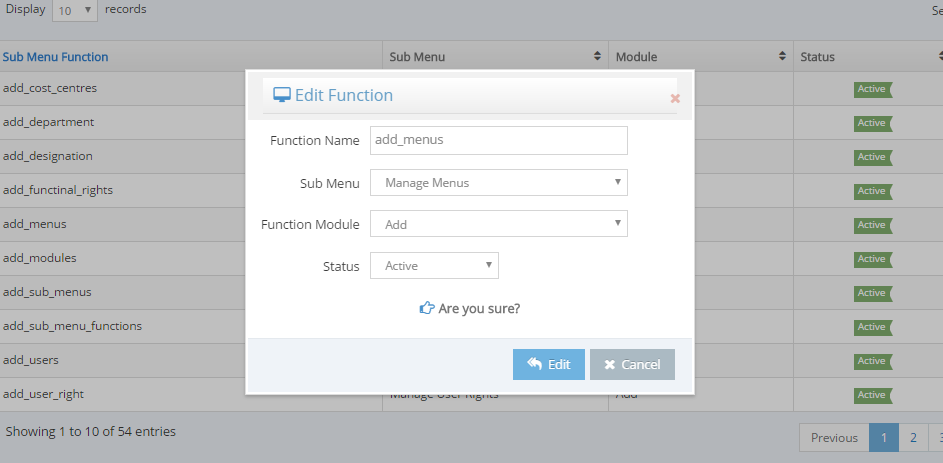
your new view : manage\_menus.php (created in application\views\ menus \)

new CSS file : manage\_menus\_css.php (created in application\views\includes\css\)

new JS file : manage\_menus\_js.php (created in application\views\includes\js\)

c. create the sub functions to the above module

steps => click "sub menu function" nav => click "add sub menu function" button => enter the data and submit.

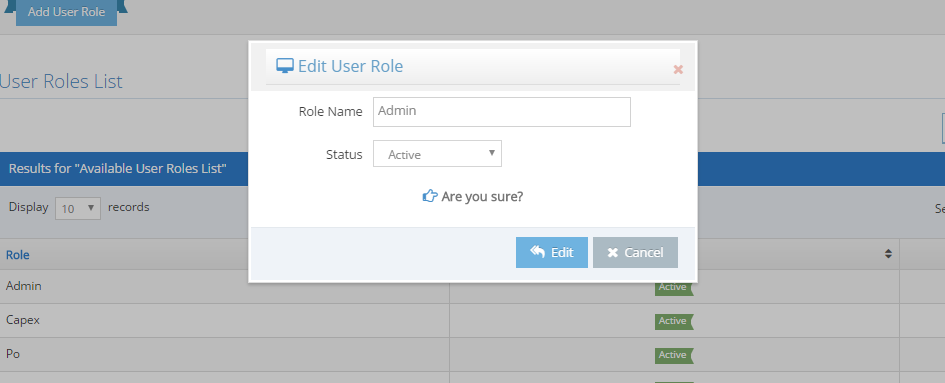
eg :- new function is "add\_menu"

so your new function is "add\_menus" in the "test" controller. you can edit the function data as you want.

How to assign these modules to the user?

1. create user roles

steps=> click "manage user roles" => click "add user role" button => add the data and submit.

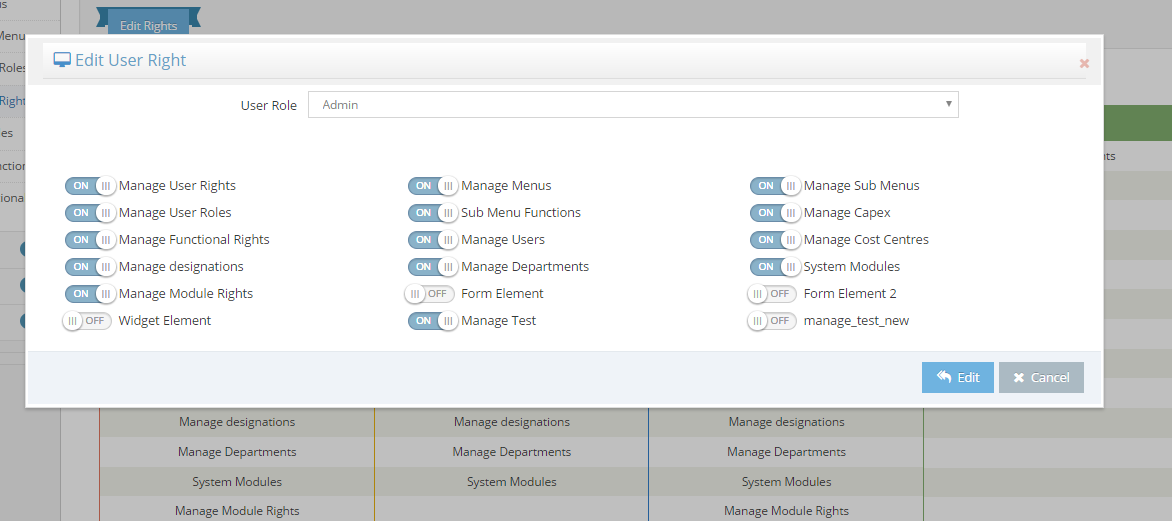
eg :- new user role id "admin"

so your new role : admin

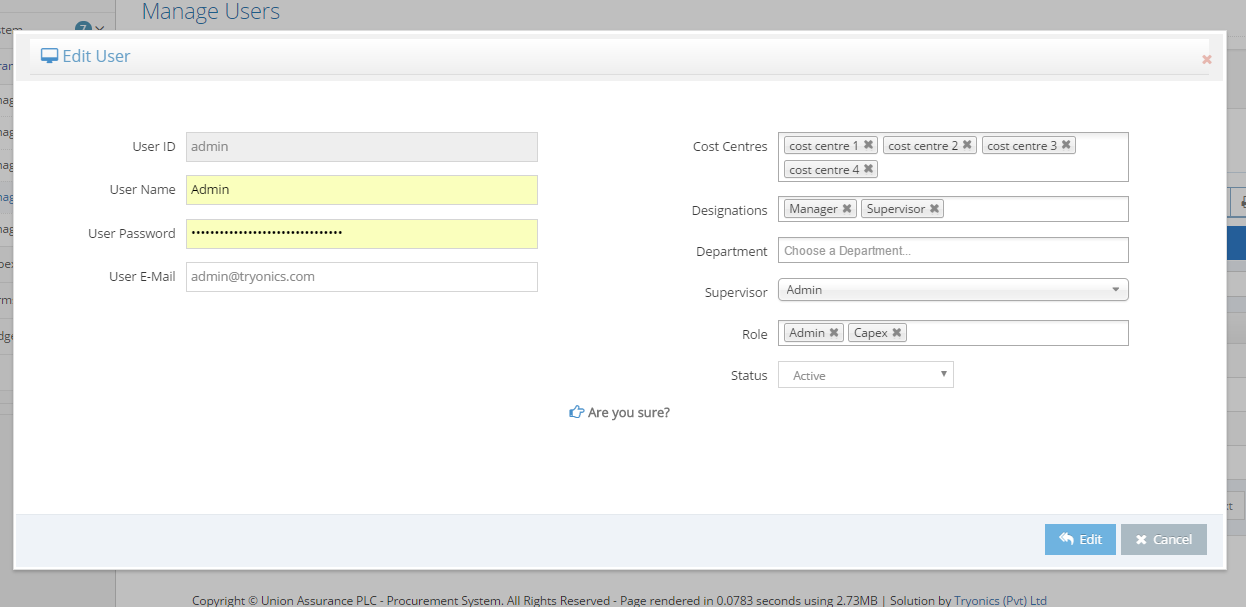
2. add the rights to the user role

steps => click "manage user rights" => click "edit rights" button => select the user role => select the "modules" which you want to give access by checking the check boxes. => click edit

this will enable the access to the modules to that user role. so users with this user roles can access that module.

eg :-

3. create a new user

steps => click "perameters" nav =>click "manage user" => click "add user " button => fill the data and submit. 

4. you can use "Manage module rights" to remove or add the function specific rights(like add/edit/view) to individual users.

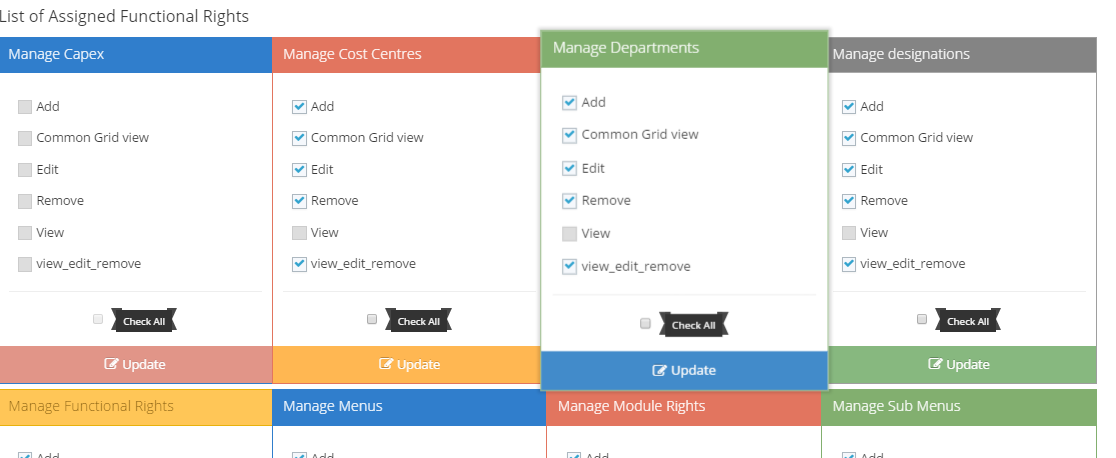
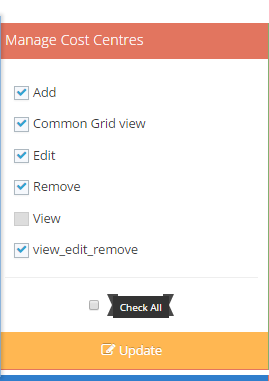
to do that ,

steps => click "pearameters" => click "manage module right" => select the "user" form user dropdown => select the "sub menu" in the sub menu dropdown => or just use the main interface to update.

=> use check all to select all rights of that module

=>if the given edit/add/view/common grid view is disabled, that means that option do not have any functions to that.

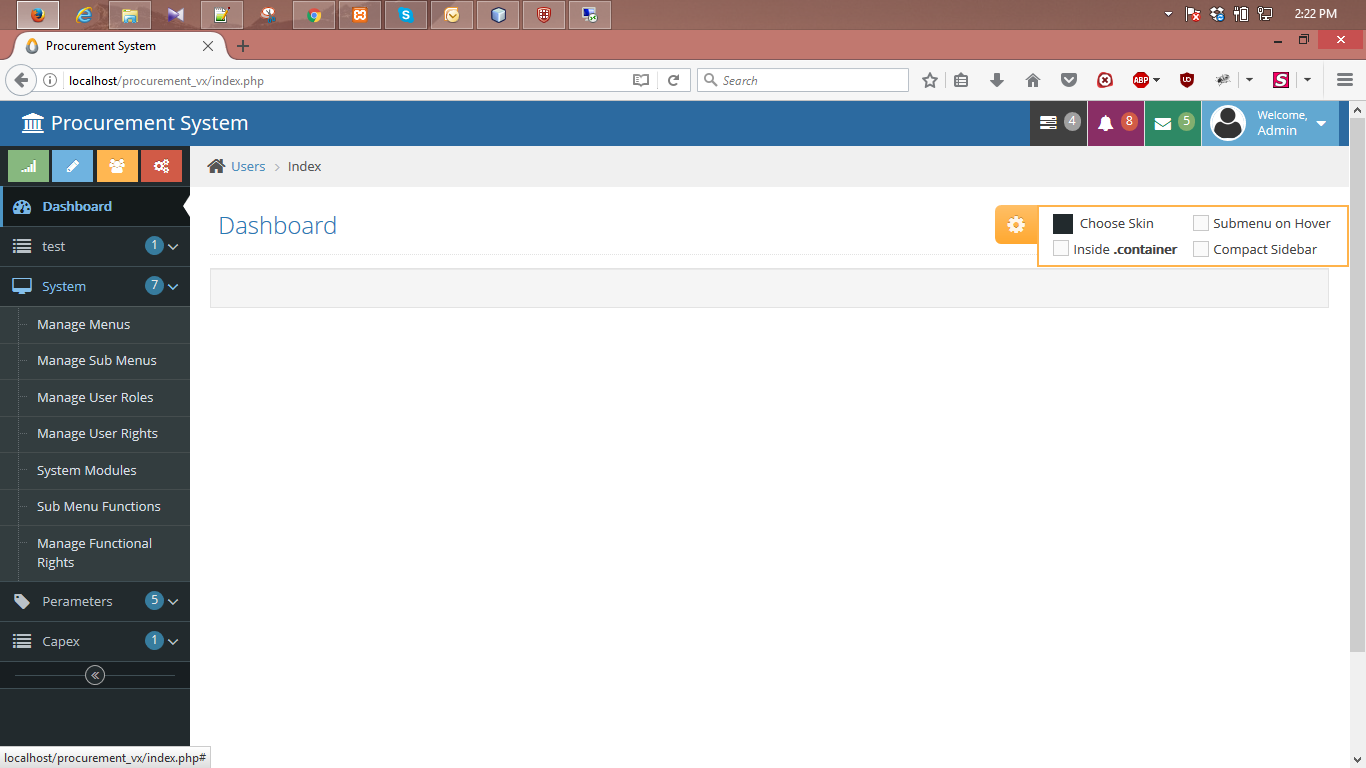
=>then click "update".

eg: - 

system template design

Codeigniter 3. 1.4 + ace-admin

user can have facility to select his own session theme setting by selecting the theme setting option.



thats all...

thank you.